

Critiques

L'idéal de co-production des savoirs est menacé d'une dépolitisation paradoxale : il situe sur un même plan la capacité de production savante des experts et des profanes alors qu'elle n'est pas socialement partagée. Dans les faits, les savoirs alternatifs aux savoirs officiels relèvent plus souvent de la contre-expertise savante que de la production profane en tant que telle. Si l'on reprend l'exemple des associations de malades, le cas des personnes homosexuelles infectées par le VIH montre bien que leur implication auprès des médecins dans la co-production des savoirs a été rendue possible par un niveau socio-éducatif et un savoir-faire militant relativement élevés⁴⁸.

Par ailleurs, l'un des fondateurs des *Social Studies of Knowledge*, Harry Collins, regrette que les savoirs experts soient mis au second rang au profit d'une vision romantique qui associe toutes les parties prenantes, alors que leur confrontation scientifique demeure prioritaire et essentielle⁴⁹. Le principe de l'expertise pluraliste est acquis, la participation des publics est maintenue comme juste, mais ceux-ci seront d'autant plus aptes à juger de l'opportunité des techniques que la confrontation scientifique des avis experts n'aura pas été négligée ni écartée. Il peut alors s'avérer utile de repérer les experts reconnus tout à la fois pour la valeur de leurs connaissances et pour leur engagement dans la société civile, susceptibles de servir de « traducteurs » au sens de Bruno Latour.

L'ingénieur informaticien et les enjeux éthiques de la conception d'outils libres

Près des trois quarts des ingénieurs informaticiens sont ingénieurs en systèmes d'information ou informaticiens de gestion, c'est-à-dire qu'ils travaillent directement sur la conception de logiciels. Un aspect essentiel de la conception de logiciels est le type de libertés qu'ils autorisent. On distingue classiquement quatre libertés fondamentales : 1) utiliser : la liberté d'utiliser le logiciel ; 2) copier : la liberté de reproduire le logiciel ; 3) étudier : la liberté d'étudier le code-source du logiciel ; 4) modifier : la liberté de modifier le code-source du logiciel. Un logiciel est dit « libre » lorsqu'il garantit ces quatre types de libertés. Les logiciels dits « propriétaires » ne permettent au contraire aucune de ces libertés (la liberté d'utilisation dépend notamment du type de licence). Entre ces deux types de logiciels, on trouve des *freewares* et des *sharewares*. Les *sharewares* sont généralement des logiciels propriétaires payants pour lesquels une version d'évaluation gratuite et limitée existe. Les *sharewares*, dits « partagiciel » ou « contribuciel », sont des logiciels qui peuvent être utilisés gratuitement, généralement durant une certaine période ou avec des fonctionnalités limitées.

Le mouvement du logiciel libre s'épanouit dans les années 1980 comme une « alternative au verrouillage par la propriété privée des savoirs et des techniques »⁵⁰. Sa figure tutélaire, Richard Stallman, programmeur et activiste du logiciel libre, déclare de manière provocatrice dans une conférence donnée en 2014 qu'il « peut [expliquer] la question du logiciel libre en trois mots : liberté, égalité, fraternité. C'est-à-dire les choses que l'État français ne respecte plus »⁵¹. On ne saurait formuler plus clairement la capacité des objets techniques à « faire de la politique ». La liberté va bien au-delà d'une simple liberté d'utilisation et repose sur la liberté de créer soi-même les conditions d'existence d'un projet en modifiant le code-source. L'égalité fait référence à une égalité d'accès dans un contexte où les promesses d'inclusion du numérique s'affaiblissent et où la fracture numérique se pare de nouvelles formes. Quant à la fraternité, elle s'exerce sur les forums Internet et les rencontres de libristes IRL – *in real life*, c'est-à-dire de façon non virtuelle, au Fosdem à Bruxelles ou lors des Samuels du libre à la Cité des sciences à Paris. Ces espaces, virtuels ou non, reposent sur le partage de compétences et le travail collaboratif dans une visée d'encapacitation (empowerment) des individus au moyen du DIY (*Do it yourself*).

Le choix du type de logiciel est un choix politique qui fait appel à deux façons radicalement différentes d'envisager la gouvernance : une approche descendante (*top-down*) dans le cas du logiciel propriétaire ou du *shareware* contre une approche ascendante (*bottom-up*) dans le cas du logiciel libre. Eric Raymond, l'un des hackers les plus célèbres de l'histoire numérique, co-créateur du terme *open source* (pour code-source ouvert), propose la métaphore de la cathédrale et du bazar pour illustrer cette différence :

« Linux a remis en cause une grande partie de ce que je croyais savoir. J'avais prêché l'évangile selon Unix sur l'utilisation de petits outils, le prototypage rapide et la programmation évolutive, depuis des années. Mais je pensais aussi qu'il existait une certaine complexité critique au-delà de laquelle une approche plus centralisée, plus a priori, était nécessaire. Je pensais que les logiciels les plus importants (comme les systèmes d'exploitation et les très gros outils comme Emacs) devaient être conçus comme des cathédrales, soigneusement élaborés par des sorciers isolés ou des petits groupes de mages travaillant à l'écart du monde, sans qu'aucune version bêta ne voie le jour avant que son heure ne soit venue. Le style de développement de Linux Torvalds – distribuez vite et souvent, déléguiez tout ce que vous pouvez déléguer, soyez ouvert jusqu'à la promiscuité – est venu comme une surprise. À l'opposé de